

Aanpassing waardebereik attribuut “stuf:functie”

In StUF 3.01 kan het attribuut "StUF:functie" de volgende waarden bevatten:¹

- entiteit
- update
- selectie
- antwoord
- zaakinfo

Deze waarden worden gebruikt om aan te geven welke functie een element in een StUF-bericht vervuld. Bijvoorbeeld in het onderstaande vrije StUF-bericht zien we twee elementen die allebei verwijzen naar het entiteitstype “Persoon” maar een andere functie vervullen, resp. “entiteit” en “update”.

```
<ns:vrij-stufbericht ...>
  <ns:stuurgegevens>...</ns:stuurgegevens>
  <ns:parameters>...</ns:parameters>
  <ns:ambtenaar stuf:entiteittype="Persoon" stuf:functie="entiteit">...</ns:ambtenaar>
  <ns:wijzigGeslachtsnaam stuf:entiteittype="Persoon" stuf:functie="update">
    <ns:parameters>
      <stuf:mutatiesoort>W</stuf:mutatiesoort>
    </ns:parameters>
    <ns:object stuf:entiteittype="Persoon" stuf:verwerkingsoort="W">...</ns:object>
    <ns:object stuf:entiteittype="Persoon" stuf:verwerkingsoort="W">...</ns:object>
  </ns:wijzigGeslachtsnaam >
</ns:vrij-stufbericht>
```

Dit is een fictief voorbeeld van een vrij bericht met daarin een naamswijziging en de ambtenaar die de wijziging heeft doorgevoerd.

Het element <ns:ambtenaar ...> representeert de entiteit “Persoon”. Het element <ns:wijzigAchternaam ...> is zelf geen entiteit maar representeert de “update” functie die wordt uitgevoerd op de entiteit “Persoon”. Binnen dit element is de body van een wijzigingskennisgeving opgenomen om de update nader te specificeren. Op deze manier kan voorgedefinieerde functionaliteit (kennisgevingen en vraag-antwoord) worden hergebruikt in vrije berichten.

¹ Zie sectie 7. 2 in het document “stuf0301.pdf”.

Voor efficiënte notatie mag het attribuut "stuf:functie" worden weggelaten als het element een entiteit is:

```
<ns:vrij-stufbericht ...>
  <ns:stuurgegevens>...</ns:stuurgegevens>
  <ns:parameters>...</ns:parameters>
  <ns:ambtenaar stuf:entiteittype="Persoon">...</ns:ambtenaar>
  <ns:wijzigAchternaam stuf:entiteittype="Persoon" stuf:functie="update">>
    <ns:parameters>
      <stuf:mutatiesoort>W</stuf:mutatiesoort>
    </ns:parameters>
    <ns:object stuf:entiteittype="Persoon" stuf:verwerkingssoort="W">...</ns:object>
    <ns:object stuf:entiteittype="Persoon" stuf:verwerkingssoort="W">...</ns:object>
  </ns:wijzigAchternaam >
</ns:vrij-stufbericht>
```

Deze verkorte notatie is bijvoorbeeld handig voor het sectormodel StUF-BRP waar alleen de functie "entiteit" gebruikt wordt.

Het eerste voorstel is om de waarde 'zaakinfo' te verwijderen uit het waardebereik van het attribuut 'stuf:functie'. Immers zaakgegevens kunnen als één of meer entiteiten uit het sectormodel StUF-ZKN in het vrije bericht worden opgenomen.

Het tweede voorstel is om het attribuut "StUF:functie" uit te breiden met de volgende waarden (op alfabetische volgorde):

- complex
- container
- gerelateerde
- groep
- onderdeel
- relatie
- simpel

Deze nieuwe waarden zijn opgenomen in het schema stuf0302.xsd dat nog in ontwikkeling is. Deze waarden maken het mogelijk om de gegevensstructuren in het content model van StUF beter te beschrijven. Hieronder zal dit worden toegelicht door middel van een aantal voorbeelden.

Gerelateerde entiteiten

In StUF 3.01 wordt op de volgende manier een relatie vanuit de entiteit Persoon naar de entiteit adres Adres gelegd.

```
<ns:persoon stuf:entiteittype="Persoon" stuf:functie="entiteit">
  <ns:bsn>123456789</ns:bsn>
  <ns:woontOp stuf:entiteittype="PersoonAdres">
    <ns:gerelateerde stuf:entiteittype="Adres">
```

```

        <adresIdentificatie>1</adresIdentificatie>
      </ns:gerelateerde>
    <aanvangVerblijf>20011203</aanvangVerblijf>
  </ns:woontOp>
</ns:persoon>

```

De relatie wordt gelegd door middel van de relatie-entiteit PersoonAdres. De entiteit Adres is hier de gerelateerde entiteit. Het nadeel van deze notatie is dat het door StUF gereserveerde element <ns:gerelateerde> niet de namespace van StUF heeft maar van het sectormodel, dus de prefix “ns” in plaats van “stuf”.

In onderstaand voorbeeld wordt dit probleem opgelost door de gerelateerde entiteit te specificeren middels het attribuut “stuf:functie” met als waarde “gerelateerde”. Het attribuut “stuf:functie” is wel netjes in de namespace van StUF gedefinieerd.

```

<ns:persoon stuf:entiteittype="Persoon" stuf:functie="entiteit">
  <ns:bsn>123456789</ns:bsn>
  <ns:woontOp stuf:entiteittype="PersoonAdres" stuf:functie="relatie">
    <adres stuf:entiteittype="Adres" stuf:functie="gerelateerde">
      <adresIdentificatie>1</adresIdentificatie>
    </adres>
    <aanvangVerblijf>20011203</aanvangVerblijf>
  </ns:woontOp>
</ns:persoon>

```

Dit is veel netter omdat we nu het begrip “gerelateerde” binnen de namespace van StUF definiëren in plaats van in de namespace (in dit geval “ns”) van een willekeurig sectormodel.

De volgende verkorte notatie waarin het attribuut “stuf:functie” op een aantal plekken is weggelaten is ook toegestaan:

```

<ns:persoon stuf:entiteittype="Persoon">
  <ns:bsn>123456789</ns:bsn>
  <ns:woontOp stuf:entiteittype="PersoonAdres" stuf:functie="relatie">
    <adres stuf:entiteittype="Adres">
      <adresIdentificatie>1</adresIdentificatie>
    </adres>
    <aanvangVerblijf>20011203</aanvangVerblijf>
  </ns:woontOp>
</ns:persoon>

```

De parser kan vrij eenvoudig afleiden dat het eerste element een “entiteit” is en dat het vierde element een “gerelateerde” is.

Impliciete relaties

Het onderstaande voorbeeld is een mogelijke variant om entiteiten te relateren waarbij de relatie impliciet is en geen eigenschappen heeft.

```

<ns:persoon stuf:entiteittype="PRS" stuf:functie="entiteit">

```

```

<ns:bsn>123456789</ns:bsn>
<adres stuf:entiteittype="ADR" stuf:functie="gerelateerde">
  <adresIdentificatie>1</adresIdentificatie>
</adres>
</ns:persoon>

```

Ook hier moet nog worden onderzocht wat de gevolgen zijn voor CRUD. De volgende verkorte notatie waarin het attribuut “stuf:functie” is weggelaten is ook toegestaan:

```

<ns:persoon stuf:entiteittype="PRS">
  <ns:bsn>123456789</ns:bsn>
  <adres stuf:entiteittype="ADR">
    <adresIdentificatie>1</adresIdentificatie>
  </adres>
</ns:persoon>

```

Dus als er binnen een element met het attribuut “stuf:entiteittype” een ander element voorkomt met hetzelfde attribuut dan is de afspraak dat het altijd een “gerelateerde” entiteit is.

Onderdelen van entiteiten

In onderstaand voorbeeld wordt aangegeven dat een entiteit onderdeel is van een andere entiteit.

```

<ns:persoon stuf:entiteittype="Tafel" stuf:functie="entiteit">
  <ns:tafel_ID>1</ns:tafel_ID>
  <ns:poot stuf:entiteittype="Poot" stuf:functie="onderdeel">
    <ns:poot_ID>1</ns:poot_ID>
  </ns:poot>
  <ns:poot stuf:entiteittype="Poot" stuf:functie="onderdeel">
    <ns:poot_ID>2</ns:poot_ID>
  </ns:poot>
  <ns:poot stuf:entiteittype="Poot" stuf:functie="onderdeel">
    <ns:poot_ID>3</ns:poot_ID>
  </ns:poot>
  <ns:poot stuf:entiteittype="Poot" stuf:functie="onderdeel">
    <ns:poot_ID>4</ns:poot_ID>
  </ns:poot>
</ns:persoon>

```

De identifiers van de onderdelen zijn lokaal en zijn pas uniek op globaal niveau als ze gecombineerd worden met de identifier van de hoofdentiteit. Dit biedt de mogelijkheid om de zogenaamde zwakke entiteiten (entiteiten die afhankelijk zijn van andere entiteiten) expliciet te benoemen. Er moet nog worden onderzocht wat de gevolgen zijn van dit nieuwe concept op de bestaande CRUD-functionaliteit van StUF.

Alleen in de (hoofd)entiteit Tafel mag het attribuut “stuf:functie” worden weggelaten. In de zwakke entiteit Poot mag dat niet omdat het anders verward kan worden met een gerelateerde entiteit van een impliciete relatie zoals beschreven in de vorige sectie.

Eigenschappen

In onderstaand voorbeeld wordt expliciet aangegeven of een eigenschap van een entiteit wel of niet simpele of complexe inhoud heeft (bijv. een getal versus GML).

```
<ns:pand stuf:entiteittype="Pand" stuf:functie="entiteit">
  <ns:identificatie stuf:functie="simpel">123456789</ns:identificatie>
  <ns:geometrie stuf:functie="complex">
    <gml:Polygon gid="p1">
      <gml:exterior>
        <gml:LinearRing>
          <gml:posList dimension="3">1. 2. 3.</gml:posList>
        </gml:LinearRing>
      </gml:exterior>
    </gml:Polygon>
  </ns:geometrie>
</ns:pand>
```

De volgende verkorte notatie is toegestaan:

```
<ns:pand stuf:entiteittype="Pand">
  <ns:identificatie>123456789</ns:identificatie>
  <ns:geometrie stuf:functie="complex">
    <gml:Polygon gid="p1">
      <gml:exterior>
        <gml:LinearRing>
          <gml:posList dimension="3">1. 2. 3.</gml:posList>
        </gml:LinearRing>
      </gml:exterior>
    </gml:Polygon>
  </ns:geometrie>
</ns:pand>
```

Let op: als een eigenschap complexe inhoud bevat is het attribuut stuf:functie="complex" altijd verplicht! Dit is nodig om complexe eigenschappen te onderscheiden van groepen in de volgende sectie.

Groepen

In onderstaand voorbeeld wordt expliciet aangegeven dat een element een groep is.

```
<ns:persoon stuf:entiteittype="Persoon" stuf:functie="entiteit">
  <ns:bsn>123456789</ns:bsn>
  <ns:geboorte stuf:functie="groep">
    <geboorteland>NL</geboorteland>
    <geboorteplaats>Rotterdam</geboorteplaats>
    <geboortedatum>20011203</geboortedatum>
  </ns:geboorte>
</ns:persoon>
```

In de huidige StUF-standaard kunnen er geen CRUD-operaties op groepen worden uitgevoerd. Bij de mGBA is daar mogelijk wel behoefte aan. Dit moet nog verder worden onderzocht.

De volgende verkorte notatie waarin het attribuut "stuf:functie" is weggelaten is ook toegestaan:

```

<ns:persoon stuf:entiteittype="Persoon">
  <ns:bsn>123456789</ns:bsn>
  <ns:geboorte>
    <ns:geboorteland>NL</ns:geboorteland>
    <ns:geboorteplaats>Rotterdam</ns:geboorteplaats>
    <ns:geboortedatum>20011203</ns:geboortedatum>
  </ns:geboorte>
</ns:persoon>

```

Het kan als volgt worden afgeleid dat het element <ns:geboorte ...> een groep voorstelt:

- het element is geen entiteit omdat het attribuut stuf:entiteittype ontbreekt
- het element is geen simpele eigenschap omdat het geneste elementen bevat
- het is geen complexe eigenschap omdat het attribuut stuf:functie="complex" ontbreekt (zie vorige sectie)

Dus de enige mogelijkheid die overblijft is dat het element een groep moet zijn.

Containers

Onderstaand voorbeeld geeft aan hoe een (betekenisloos) strikje kan worden gelegd om entiteiten te groeperen.

```

<ns:personen stuf:functie="container">
  <ns:persoon stuf:entiteittype="Persoon" stuf:functie="entiteit">
    <ns:bsn>123456789</ns:bsn>
    <ns:geboorte stuf:functie="groep">
      <geboorteland>NL</geboorteland>
      <geboorteplaats>Rotterdam</geboorteplaats>
      <geboortedatum>20011203</geboortedatum>
    </ns:geboorte>
  </ns:persoon>
  <ns:persoon stuf:entiteittype="Persoon" stuf:functie="entiteit">
    <ns:bsn>223456789</ns:bsn>
  </ns:persoon>
  <ns:persoon stuf:entiteittype="Persoon" stuf:functie="entiteit">
    <ns:bsn>323456789</ns:bsn>
  </ns:persoon>
</ns:personen>

```

Een container is vooralsnog anders dan een groep. Containers worden gebruikt om entiteiten te clusteren en groepen worden gebruikt om eigenschappen en relaties van entiteiten te groeperen. Het is wellicht ook mogelijk om het concept "groep" in StUF te verruimen zodat je met een groep ook entiteiten kunt groeperen. Echter dit vraagt om aanpassing van een bestaand concept.

De volgende verkorte notatie waarin het attribuut "stuf:functie" is weggelaten is ook toegestaan in de nieuwe StUF 3.02 syntax:

```

<ns:personen>
  <ns:persoon stuf:entiteittype="Persoon">

```

```

    <ns:bsn>123456789</ns:bsn>
    <ns:geboorte>
      <geboorteland>NL</geboorteland>
      <geboorteplaats>Rotterdam</geboorteplaats>
      <geboortedatum>20011203</geboortedatum>
    </ns:geboorte>
  </ns:persoon>
  <ns:persoon stuf:entiteittype="Persoon">
    <ns:bsn>223456789</ns:bsn>
  </ns:persoon>
  <ns:persoon stuf:entiteittype="Persoon">
    <ns:bsn>323456789</ns:bsn>
  </ns:persoon>
</ns:personen>

```

Als een element zelf geen entiteit is, maar wel een child-element bevat dat wel entiteit is, dan is het parent-element een container.